
A Low Rank Approach to Automatic Differentiation

Hany S. Abdel-Khalik¹, Paul D. Hovland^{2,3}, Andrew Lyons³, Tracy E. Stover¹, and Jean Utke^{2,3}

¹ Department of Nuclear Engineering, North Carolina State University, Raleigh, NC 27695-7909 abdelkhalik@ncsu.edu, testover@ncsu.edu

² Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Ave., Argonne, IL 60439, hovland@mcs.anl.gov

³ Computation Institute, University of Chicago, 5640 S. Ellis Avenue, Chicago, IL 60637 lyonsam@gmail.com, utke@mcs.anl.gov

Summary. This manuscript introduces a new approach for increasing the efficiency of automatic differentiation (AD) computations for estimating the first order derivatives comprising the Jacobian matrix of a complex large-scale computational model. The objective is to approximate the entire Jacobian matrix with minimized computational and storage resources. This is achieved by finding low rank approximations to a Jacobian matrix via the Efficient Subspace Method (ESM). Low rank Jacobian matrices arise in many of today’s important scientific and engineering problems, e.g. nuclear reactor calculations, weather climate modeling, geophysical applications, etc. A low rank approximation replaces the original Jacobian matrix J (whose size is dictated by the size of the input and output data streams) with matrices of much smaller dimensions (determined by the numerical rank of the Jacobian matrix). This process reveals the rank of the Jacobian matrix and can be obtained by ESM via a series of r randomized matrix-vector products of the form: $J\mathbf{q}$, and $J^T\omega$ which can be evaluated by the AD forward and reverse modes, respectively.

Key words: Forward mode, reverse mode, efficient subspace method, low rank

1 Introduction

AD has arisen as a powerful tool that can potentially meet the need for efficient and accurate evaluation of sensitivity information, i.e. derivatives, for complex engineering models. Derivative information is required for a wide range of engineering and research-oriented tasks, e.g. design optimization, code-based uncertainty propagation, and data assimilation.

The functionality of AD depends to a large extent on the complexity of the engineering model to be differentiated. With the startling growth in computer power, and the implementation of efficient computer algorithms, the application of AD to realistic engineering models has been made feasible [5, 9]. In many of today’s complex engineering systems, e.g. modeling of nuclear phenomena, weather climate modeling, however, it is safe to say that most of the associated computational models, either deterministic and/or probabilistic, operate at the limit of the computing capacity of the state-of-the-art computing resources. Therefore, it is

paramount to increase the efficiency of AD algorithms to a level that enables their application to complex large-scale engineering models.

The efficiency of AD can be increased depending on the type of the problem, and the sparsity pattern of the Jacobian matrix. For example, if the number of input data is relatively small, and number of output data is large, the forward mode of differentiation presents the most efficient way with regard to computational time and storage burdens. Conversely, the reverse mode of differentiation suits problems with many input data and few output data. If the Jacobian matrix is sparse, one can propagate sparse derivative vectors [4] or compress the Jacobian using coloring techniques [3, 7].

This manuscript addresses the need for a general approach when both the numbers of input and output data are too large to render either the forward and/or the reverse modes computationally feasible, and when the Jacobian matrix is generally dense. Proposed is a new approach that utilizes the Efficient Subspace Method (ESM) to address these situations [2]. The sole requirement for this approach is that the Jacobian matrix be ill-conditioned, which is generally considered an unfavorable situation. ESM exploits the ill-conditioning of the Jacobian matrix to reduce the number of runs of the forward and reverse modes to a minimum. We will show that the number of runs is proportional to the numerical rank of the Jacobian matrix which is determined as part of the analysis. In this approach, the Jacobian matrix of the engineering model with m output data and n input data is approximated by matrices of lower dimensions by means of matrix revealing decompositions. These decompositions are obtained by ESM via a series of randomized matrix-vector products of the form: $J\mathbf{q}$, and $J^T\boldsymbol{\omega}$, where r is the *numerical rank* of the Jacobian matrix J . Note that the size of the Jacobian matrix is dictated by the size of the input and output data streams, however the sizes of the smaller matrices characterizing the low rank decomposition are determined by the numerical rank of the Jacobian matrix, which is found to be related to the modeling strategy, physics of the engineering system, and the degree of correlation amongst the input data, which can be quite significant. This follows, since for many important engineering applications, the input data to a computational model are the output from other preprocessor codes.

Earlier work by the first author has demonstrated that for typical nuclear reactor models, the numerical rank is many orders of magnitude smaller than the size of the input and output data streams, i.e. $r \ll m, n$ [8], and can be estimated effectively via ESM (for typical nuclear reactor core simulation: $n \approx 10^6$, $m \approx 10^5$, and $r \approx 10^2$). This large rank reduction is a result of the so-called multi-scale phenomena modeling (MSP) strategy on which nuclear reactor calculations are based. Nuclear calculations are but an example of the application of MSP to engineering systems that involve large variations in both time and length scales. In fact, many of today's important engineering and physical phenomena are modeled via MSP, e.g. weather forecast, geophysics, materials simulation. To accurately model the large time and scale variations, MSP utilizes a series of models varying in complexity and dimensionality. First, high resolution (HR) microscopic models are employed to capture the basic physics and the short scales that govern system behavior. The HR models are then coupled with low resolution (LR) macroscopic models to directly calculate the macroscopic system behavior, which is often of interest to system designers, operators, and experimentalists. The coupling between the different models results in a gradual reduction in problem dimensionality thus creating large degrees of correlations between different data in the input and output (I/O) data streams. ESM exploits this situation by treating the I/O data in a collective manner in search of the independent pieces of information. The term 'Degree of Freedom' (DOF), adopted in many other engineering fields, is used to denote an independent piece of information in the I/O stream. An active DOF denotes a DOF that is transferred from a higher to a lower resolution model, and an inactive DOF denotes a DOF that is thrown out. ESM replaces the original I/O

data streams by their corresponding active DOFs. The number of active DOFs can be related to the numerical rank of the Jacobian matrix. The reader is referred to a previous publication for a full treatment of ESM theory [2].

2 Methodology

Let the engineering model of interest be described by a vector valued function:

$$\mathbf{y} = \Theta(\mathbf{x}) \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{x} \in \mathbb{R}^n$. The objective of this manuscript is to minimize the computational and storage overheads required to calculate the entire Jacobian matrix J to a prescribed error tolerance limit. The elements of the Jacobian matrix contain derivative information that is given by:

$$J_{ij} = \frac{\partial y_i}{\partial x_j} \quad (2)$$

Eq. 2 describes the sensitivity of the i th output response with respect to the j th input model parameter. For typical nuclear reactor calculations, earlier work has revealed that a) model input data (also referred to as model input parameters) number in the millions. For example, the few-group cross-sections (cross-sections characterize the probability of neutrons interaction with matter), input to a core simulator are often functionalized in terms of history effects, and various instantaneous core conditions thus result in a very large input data stream, and b) model output responses number in the hundreds of thousands, including in-core instrumentations' responses, core power distribution, and various thermal limits, thus resulting in a very large output data stream. The numerical rank of the associated Jacobian matrix has been shown to be much smaller, i.e. of the order of 10^2 only. A low rank matrix suggests a matrix revealing decomposition of the form:

$$J = USV^T \quad (3)$$

where $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{n \times r}$, and $S \in \mathbb{R}^{r \times r}$. To simplify the derivation, we select S to be diagonal, and both U and V orthonormal, thus yielding the singular value decomposition (SVD) of the matrix J . Note that the columns of U span a subspace of dimension r that belongs to the output responses space of dimension m , i.e. $R(U) \in \mathbb{R}^m$, and $\dim(R(U)) = r$, where $R(\cdot)$ denotes the range of a matrix operator, and $\dim(\cdot)$ is the dimension of a subspace which represents the maximum number of linearly independent vectors that belong to a subspace. Similarly: $R(V) \in \mathbb{R}^n$, and $\dim(R(V)) = r$.

Two important observations can be made about the decomposition in Eq. 3: (a) for any vector \mathbf{v} , such that: $V^T \mathbf{v} = \mathbf{0}$, i.e. \mathbf{v} is orthogonal to the r columns of the matrix V , the following condition is true: $J\mathbf{v} = \mathbf{0}$, i.e. a change of model input parameters along the vector \mathbf{v} does not lead to any change in model output responses, i.e. the sensitivity of model responses with respect to the direction \mathbf{v} is zero. In other words, this vector matrix product carries no derivative information and hence need not be evaluated. In our notations, any vector satisfying this condition is called an input data inactive DOF. For a matrix with rank r , there are $n - r$ inactive DOFs. (b) Similarly, for any vector \mathbf{u} , such that: $U^T \mathbf{u} = \mathbf{0}$, the following condition is satisfied: $J^T \mathbf{u} = \mathbf{0}$. Again, this matrix-transpose-vector product produces the null vector and hence need not be evaluated. There are $m - r$ output data inactive DOFs.

Based on these two observations, it is useful to seek an approach that evaluates the effect of the matrix J on the active input and output data DOFs only. Clearly, this approach is challenged

by the lack of knowledge of the matrices U and V . These matrices can only be calculated if the matrix J is available a priori. ESM can approximate these matrices by using a series of r matrix-vector and matrix-transpose-vector products only.

The mechanics of ESM are based on the following three theorems:

Theorem 1. *Let $J \in \mathbb{R}^{m \times n}$ be a pre-defined matrix (representing an unknown Jacobian matrix of a computer code) with rank $r \leq \min(m, n)$. Let $J = URV^T$ be a matrix revealing decomposition as described earlier. Given $Q \in \mathbb{R}^{n \times l}$ a matrix of randomly generated entries, then Q has a unique decomposition such that:*

$$Q = Q^P + Q^\perp,$$

where

$$\text{rank}(Q^P) = l$$

and

$$\begin{aligned} R(Q^P) &\subseteq R(V) \text{ for } l \leq r \\ R(Q^P) &= R(V) \text{ for } l > r \end{aligned}$$

Theorem 2. *Let $D = JQ$ (action of AD forward mode), then:*

$$\text{rank}(D) = l,$$

and

$$\begin{aligned} R(D) &\subseteq R(U) \text{ for } l \leq r \\ R(D) &= R(U) \text{ for } l > r \end{aligned}$$

Theorem 3. *Given $\tilde{U} \in \mathbb{R}^{m \times l}$, a matrix of randomly generated entries, and an arbitrary matrix J^* such that: $R(J^*) = R(V)$, let $Z = J^* \tilde{U}$, $Z \in \mathbb{R}^{n \times l}$ then:*

$$\text{rank}(Z) = l,$$

and

$$\begin{aligned} R(Z) &\subseteq R(V) \text{ for } l \leq r \\ R(Z) &= R(V) \text{ for } l > r \end{aligned}$$

These theorems guarantee that one can gradually build a basis for the subspaces comprising the active DOFs in the input and the output data streams, i.e. $R(V)$, and $R(U)$, respectively. Further, theorem 3 provides enough flexibility in the choice of the operator J^* used to build the input data active DOFs subspace, $R(V)$. A simple choice is $J^* = J^T$ which reduces to the direct implementation of AD reverse mode. As will be described later, this flexibility will suggest means of reducing the computational overhead required to execute the AD reverse mode, which is often more expensive than the AD forward mode.

Instead of reproducing the proofs for these theorems which may be found elsewhere [1], it will be instructive to illustrate the mechanics of these theorems by introducing a few simplifying assumptions that will be relaxed later. Let us assume that the rank r and the subspace $R(V)$

associated with the Jacobian matrix J are known a priori. Let the subspace $R(V)$ be spanned by the columns of an orthonormal matrix Q , such that: $Q = [\mathbf{q}_1 \mathbf{q}_2 \dots \mathbf{q}_r]$, $\mathbf{q}_i^T \mathbf{q}_j = \delta_{ij}$, and $R(V) = R(Q)$. Note that only the subspace $R(V)$ is assumed known; identifying the columns of the matrix V represents the target of our analysis. Now, since Q is orthonormal and its columns span $R(V)$, one can write:

$$Q = VP^T \quad (4)$$

where $P \in \mathbb{R}^{r \times r}$ is a full rank orthonormal matrix, also known as a rotation operator, i.e. it rotates the columns of the matrix Q to line up with the columns of the matrix V . In this analysis, Q is selected arbitrarily, therefore V can be extracted from Eq. 4, once P is determined, according to: $V = QP$. Now, assuming that Eq. 1 is implemented as a computer program and AD has been applied yielding a differentiated program with a capability to calculate matrix-vector products, one can evaluate the matrix D of output responses:

$$D = JQ = [J\mathbf{q}_1 J\mathbf{q}_2 \dots J\mathbf{q}_r] \quad (5)$$

Substituting for J and Q from Eq. 3 and Eq. 4 yields:

$$D = USP^T. \quad (6)$$

This is the SVD of the matrix D . Hence, if one calculates the SVD of the matrix D , one can reconstruct the matrix J as follows:

$$J = (DP)(QP)^T = USV^T. \quad (7)$$

Therefore, one can evaluate the entire Jacobian matrix with only r matrix-vector products (Eq. 5) and an SVD operation for a matrix of r columns only (Eq. 6), given that one knows a priori the subspace $R(V)$ and its dimension r .

Now we turn to relaxing these two assumptions. According to Theorem 3, the subspace $R(V)$ may be identified by using the reverse differentiation mode of AD, by making a simple choice of J^* such that: $J^* = J^T$. In this case, a basis for the $R(V)$ subspace can be constructed in the following manner: a) Build a matrix \tilde{U} of randomly generated entries, where $l \leq r$, and perform the following matrix-vector products using the AD reverse mode:

$$Z = J^T \tilde{U}$$

b) Calculate a QR decomposition of the matrix Z :

$$Z = QR$$

where $Q \in \mathbb{R}^{m \times l}$ is an orthonormal matrix of rank l , such that: $R(Q) = R(Z)$. For a random \tilde{U} , theorem 3 asserts that:

$$R(Q) \subseteq R(J^T) = R(V),$$

and for $l > r$

$$R(Q) = R(V).$$

Therefore, one can build a low rank approximation to a Jacobian matrix by: a) First, using the AD reverse mode to construct a basis for the subspace $R(V)$. This subspace is constructed by performing matrix-transpose-vector products until the entire subspace is spanned. b) Second, using the AD forward mode, one can identify the three matrices of the SVD in Eq. (2). Once SVD is calculated, one can estimate the J_{ij} according to:

$$J_{ij} = \sum_{k=1}^r u_{ik} s_k v_{jk}.$$

Now we would like to comment on the choice of the matrix J^* . As illustrated above, the primary function of J^* is to construct an arbitrary basis for the subspace $R(V)$; once $R(V)$ is identified, the forward mode can be used to estimate the entire Jacobian matrix. Therefore, any matrix that has the same range as the matrix J^T can be used to build $R(V)$. Having this insight can help one make an educated choice of the matrix J^* for one's particular application of interest. To illustrate this: consider that, in most computer codes, the Jacobian matrix J comprises a series of calculational stages, where the output of one stage feeds the input to the next stage. Mathematically, this can be described using the chain rule of differentiation:

$$J = J_1 J_2 \dots J_N$$

And the transposed Jacobian matrix becomes:

$$J^T = J_N^T \dots J_2^T J_1^T$$

Now, each J_i represents the Jacobian of a calculational stage, and $\{J_i\}$ are generally expected to vary in dimensionality, sparsity pattern, and numerical rank. One can take advantage of this situation by dropping all matrices that do not contribute to reducing the overall rank of the Jacobian matrix. This follows since the range of the overall transposed-Jacobian matrix satisfies the following relation:

$$R(J^T) = R(J_N^T \dots J_2^T J_1^T) \subseteq \dots \subseteq R(J_2^T J_1^T) \subseteq R(J_1^T)$$

Therefore, if say, the rank of the matrix J_1^T is comparable to the overall rank of the Jacobian matrix (ranks can be determined effectively via AD forward mode as guaranteed by theorem 2), one can use $J^* = J_1^T$ and save the additional effort required to build a reverse AD mode for the entire Jacobian matrix.

3 Case Study

This section describes a numerical experiment conducted to illustrate the mechanics of the proposed approach. Consider the diffusion equation for mono-energetic neutrons in two-dimensional non-multiplying media:

$$-\nabla \cdot D(\mathbf{r}) \nabla \Phi(\mathbf{r}) + \Sigma_a(\mathbf{r}) \Phi(\mathbf{r}) = S(\mathbf{r}) \quad (8)$$

where input model parameters are the diffusion coefficient, D , the absorption cross-section, Σ_a , and the external neutron source, S ; and \mathbf{r} denotes model parameters' variations with space. The solution to this equation gives the neutron flux, Φ .

This problem can be solved using a Galerkin formulation of the finite element method [10]. In this method, the flux solution is expanded along a basis of finite dimensional subspace of an admissible Hilbert space, denoted by the 'solution subspace'. Further, the residual error resulting from this approximation is required to be orthogonal to another Hilbert subspace, denoted by the 'residual subspace'. Usually, the two subspaces are selected to coincide with each other. Mathematically, the flux solution $\Phi^G(\mathbf{r})$ calculated by the Galerkin approach may be described as follows. Let the Galerkin flux solution and residual be given by:

$$\Phi^G(\mathbf{r}) = \psi_0 + \sum_{g=1}^G \psi_g(\mathbf{r}) \quad (9)$$

$$\varepsilon(\mathbf{r}) = S(\mathbf{r}) - \Sigma_a(\mathbf{r}) \Phi^G(\mathbf{r}) + \nabla \cdot D(\mathbf{r}) \nabla \Phi^G(\mathbf{r}) \quad (10)$$

where the solution and residual subspaces are spanned by the set of functions, $\psi_g(\mathbf{r})|_{g=0}^G$, and $G+1$ is the dimension of each of these subspaces. The following condition is also satisfied:

$$\langle \varepsilon(\mathbf{r}), \psi_g(\mathbf{r}) \rangle = 0, \quad g = 0, 1, \dots, G$$

where $\langle \cdot, \cdot \rangle$ denotes inner product over the phase space. Finally, some boundary conditions are imposed to get closure relations:

$$\Phi^G(\mathbf{r}) \Big|_{\mathbf{r} \in B} = Y(\mathbf{r}) \quad (11)$$

where $Y(\mathbf{r})$ is a function defined on the boundary B .

For typical nuclear reactor calculations, Eq. 8 through Eq. 11 are solved over a spatial grid that spans the entire reactor core. In this regard, a typical sensitivity study would involve the estimation of the Jacobian matrix relating first order changes in the flux solution to variations in input model parameters.

For this problem, a mesh size of $N = 10$ in both the x - and y -directions was selected, yielding a total of $N^2 = 100$ mesh points. The total number of input model parameters, including diffusion coefficient, absorption cross-section and source term, is $3N^2$, each parameter evaluated at N^2 grid points. The total number of output responses is N^2 , representing flux solution at the same number of grid points. Therefore the Jacobian matrix is expected to be of dimensions: $J \in \mathbb{R}^{N^2 \times 3N^2}$. To construct the entire Jacobian matrix, the direct forward and reverse AD modes of differentiation will require $3N^2$ and N^2 model evaluations, respectively. For complex nuclear calculations, the computing times required by such model re-evaluations severely restrict the scope of sensitivity analysis. In practice, the core designer is restricted to perform the sensitivity study for a few number of output responses, i.e. flux solutions at few grid points, and a few number of input parameters that are judged to be of most importance.

We selected the basis functions spanning the ‘solution subspace’ and the ‘residual subspace’ such that

$$\psi_g(x, y) = f_l(x) \times f_k(y),$$

where $l = 1, \dots, L$; $k = 1, \dots, K$; $g = 0, 1, \dots, (l-1)K + k, \dots, LK$; $f_l(x)$ and $f_k(y)$ are polynomials of order l and k , respectively; and ψ_0 is a constant function. These basis functions are often selected to satisfy special orthogonality properties to facilitate the process of obtaining the flux solution. For more details on the constructions of these basis functions, the reader is referred to the relevant literature [6]. For this study, we selected $L = K = 4$, i.e. a total of 4 polynomials in each direction and a constant term, the dimension of the ‘solution subspace’ is $G+1 = 17$. Therefore, the rank of the Jacobian matrix is expected to be 17 as well. This follows, since all possible neutron flux variations resulting from input model parameters variations must belong to the ‘solution subspace’.

Accordingly, the approach proposed in this paper, Eq. 4 through Eq. 7, was implemented by gradually increasing the size of the random subspace until $r = 17$, above which the rank did not increase. The Jacobian matrix was constructed both using the proposed approach with r forward and r reverse model evaluations using AD. In addition, a full sensitivity study was performed by running the forward mode $3N^2$ times to construct the entire Jacobian matrix. Figures 1 through 4 plot the variations of flux solution with respect to 1% perturbation in the absorption cross-section, and the source term at two random grid points. In these figures, the nodes indices are in the natural order, i.e. $q = i + (j-1) \times I$. Each of these figures compares the AD forward mode obtained by $3N^2$ model re-evaluations, and the proposed approach with only $2r$ model re-evaluations.

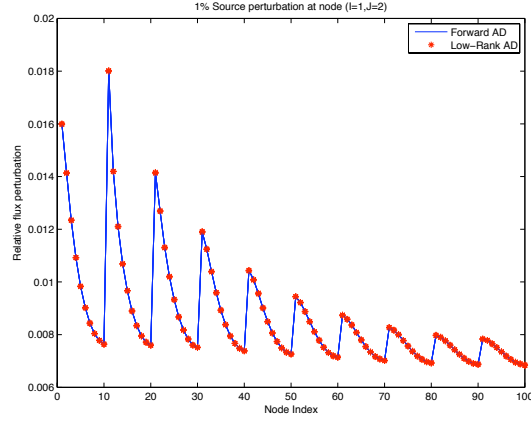


Fig. 1. First order flux perturbation due to source perturbations

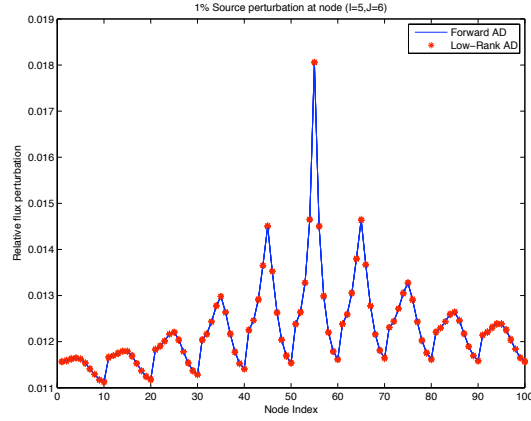


Fig. 2. First order flux perturbation due to source perturbations

4 Conclusions and Future Work

This work proposed a new approach to increase the efficiency of automatic differentiation calculations by exploiting the low rank nature of the Jacobian matrices often encountered with most very large and complex computer models. The approach requires r matrix-transpose-vector products evaluated by the AD reverse mode, r matrix-vector products evaluated by the AD forward mode, and a QR and an SVD factorization both involving matrices with r columns only. The full Jacobian matrix may subsequently be calculated from the resulting QR and SVD matrices. For an exactly rank-deficient Jacobian matrix with rank r , the proposed approach guarantees that the reconstructed Jacobian matrix is exactly equal to the full Jacobian matrix

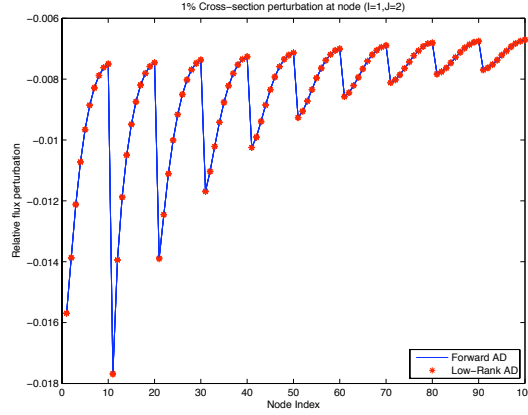


Fig. 3. First order flux perturbation due to absorption cross-section perturbations

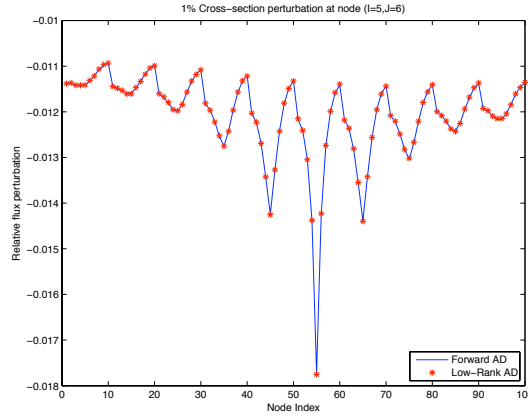


Fig. 4. First order flux perturbation due to absorption cross-section perturbations

evaluated using AD with n forward mode runs, or m reverse mode runs, where n and m are the number of input and output data, respectively.

In this work, the rank of the Jacobian matrix was upper-bounded by the size of the Galerkin's 'solution subspace' implying that the Jacobian matrix is exactly rank-deficient, and that is it has a finite number of non-zero singular values, with all the rest equal to zero. In more general situations, all the singular values of the Jacobian matrix are not exactly equal to zero, however they decrease rapidly to very small values. An engineering judgment must be made regarding the appropriate cut-off on the spectrum of singular values in order to yield an acceptable estimate for the derivatives. Future work will focus on mathematically quantifying the error in estimating the Jacobian matrix when the smaller singular values assumed to have

value zero. Further, we will extend the proposed methodology to estimating the higher order derivatives for nonlinear computer models.

Acknowledgement. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.

References

1. Abdel-Khalik, H.S.: Adaptive core simulation. Ph.D. thesis, North Carolina State University (2004)
2. Abdel-Khalik, H.S., Turinsky, P.J., Jessee, M.A.: Efficient subspace methods-based algorithms for performing sensitivity, uncertainty, and adaptive simulation of large-scale computational models (2007)
3. Averick, B.M., Moré, J.J., Bischof, C.H., Carle, A., Griewank, A.: Computing large sparse Jacobian matrices using automatic differentiation. *SIAM J. Sci. Comput.* **15**(2), 285–294 (1994)
4. Bischof, C.H., Khademi, P.M., Bouaricha, A., Carle, A.: Efficient computation of gradients and jacobians by dynamic exploitation of sparsity in automatic differentiation. *Optimization Methods and Software* **7**(1), 1–39 (1996). DOI 10.1080/10556789608805642
5. Bücker, H.M., Lang, B., Rasch, A., Bischof, C.H.: Computation of sensitivity information for aircraft design by automatic differentiation. In: P.M.A. Sloot, C.J.K. Tan, J.J. Dongarra, A.G. Hoekstra (eds.) *Computational Science – ICCS 2002, Proceedings of the International Conference on Computational Science, Amsterdam, The Netherlands, April 21–24, 2002. Part II, Lecture Notes in Computer Science*, vol. 2330, pp. 1069–1076. Springer, Berlin (2002)
6. Finnemann, H., Bennewitz, F., Wagner, M.: Interface current techniques for multi-dimensional reactor calculations. *Atomkernenergie (ATKE)* **30**, 123–128 (1977)
7. Gebremedhin, A.H., Manne, F., Pothen, A.: What color is your jacobian? graph coloring for computing derivatives. *SIAM Review* **47**(4), 629–705 (2005). DOI 10.1137/S0036144504444711. URL <http://link.aip.org/link/?SIR/47/629/1>
8. Jessee, M.A., Abdel-Khalik, H.S., Turinsky, P.J.: Evaluation of BWR core attributes uncertainties due to multi-group cross-section uncertainties. In: *Joint International Meeting on Mathematics and Computation, and Supercomputing in Nuclear Applications* (2007)
9. Losch, M., Heimbach, P.: Adjoint Sensitivity of an Ocean General Circulation Model to Bottom Topography. to appear in *Journal Of Physical Oceanography* (2006)
10. Zienkiewicz, O., Taylor, R.: *The Finite Element Method*, fourth edition edn. McGraw-Hill, New York (1989)

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.